

## Claims

[c1] What is claimed is: 1. A method for a novel model for dynamic server-side generation of computer program code which can be executed at the client machine, the method comprising the steps of:  
Having an application consisting of three main sections, which subdivide the application into layers of functionality, the Modules section, the Options section and the Content section;  
Having an Application Logic Engine; and  
Having a plurality of server and client computing means.

[c2] 2. The method of claim 1 in which said Modules section includes all mutually-exclusive components of said application.

[c3] 3. The method of claim 1 in which said Options section includes all options which implement the functionality of every Module.

[c4] 4. The method of claim 1 in which said Options section can contain a plurality of sub-options.

[c5] 5. The method of claim 1 in which said Content section comprises all the actual functionality of said Application.

[c6] 6. The method of claim 1 in which said Application Logic Engine has the following interconnected components:  
i) An Application Building Engine that functions as a core arbiter that controls and directs the construction of applications and serves as link between all Application Logic Engine modules and other parts of the system;  
ii) An Application Descriptor Repository that serves as a data repository that contains details about the application structure, including its modules, options, suboptions and content;  
iii) An Application Descriptor Interface that interfaces and acts as an abstraction layer between the Application Building Engine and the Application Descriptor Repository;  
iv) An Application Implementation Reference Repository is a data repository that contains client-specific implementation details of an application;

v) An Application Implementation Reference Interface that interfaces the Application Implementation Reference Repository and Application Building Engine; and

vi) An Application Logic Engine Interconnection Pipe which interconnects a plurality of Application Logic Engines.

[c7] 7. The method of claim 1 in which includes the following step:

Having server means create customized applications in response to client means request.

[c8] 8. The method of claim 1 in which the Application Logic Engine is integrated into an other software component architecture model.

[c9] 9. The method of claim 1 in which includes the following steps:

Posting of a request from a client means to a server means;

Examining the request by the server means;

Passing the request to the Application Logic Engine on the Server means;

Having said request contain an application identifier and functionality

information; Having said application identifier refering to the type of service in the request;

Having said Server implement required functionality in a Client-specific language and fulfill the request;

Retrieving all application structure details including modules, options and suboptions from a data repository;

Retrieving the required platform and environment information that implements the required functionality;

Interfacing the application structures and required functionality; and

Delivering the finished application or an error message if application does not implement all logical descriptors required by the client means application request.

[c10] 10. A computer program product for a novel model for dynamic server-side generation of computer program code which can be executed at the client machine, the computer program product comprising a computer usable medium having computer readable program code thereon, including:

Computer code for having an application consisting of three main sections, which subdivide the application into layers of functionality, the Modules section, the Options section and the Content section;

Computer code for having an Application Logic Engine; and

Having a plurality of server and client computing means.

[c11] 11. The computer program product of claim 10 in which said Modules section includes all mutually-exclusive components of said application.

[c12] 12. The computer program product of claim 10 in which said Options section includes all options which implement the functionality of every Module.

[c13] 13. The computer program product of claim 10 in which said Options section can contain a plurality of sub-options.

[c14] 14. The computer program product of claim 10 in which said Content section comprises all the actual functionality of said Application.

[c15] 15. The computer program product of claim 10 in which said Application Logic Engine has the following interconnected components:

i) An Application Building Engine that functions as a core arbiter that controls and directs the construction of applications and serves as link between all Application Logic Engine modules and other parts of the system;

ii) An Application Descriptor Repository that serves as a data repository that contains details about the application structure, including its modules, options, suboptions and content;

iii) An Application Descriptor Interface that interfaces and acts as an abstraction layer between the Application Building Engine and the Application Descriptor Repository;

iv) An Application Implementation Reference Repository is a data repository that contains client-specific implementation details of an application;

v) An Application Implementation Reference Interface that interfaces the Application Implementation Reference Repository and Application Building Engine; and

vi) An Application Logic Engine Interconnection Pipe which interconnects a

plurality of Application Logic Engines.

[c16] 16. The computer program product of claim 10 in which includes the following step:

Having server means create customized applications in response to client means request.

[c17] 17. The computer program product of claim 10 in which the Application Logic Engine is integrated into an other software component architecture model.

[c18] 18. A computer program to add a new terminal device to a system using the computer program in claim 10 with interfaces and the program code for used for:

Posting of a request from a client means to a server means;

Examining the request by the server means;

Passing the request to the Application Logic Engine on the Server means;

Having said request contain an application identifier and functionality information; Having said application identifier refering to the type of service in the request;

Having said Server implement required functionality in a Client-specific language and fulfill the request;

Retrieving all application structure details including modules, options and suboptions from a data repository;

Retrieving the required platform and environment information that implements the required functionality;

Interfacing the application structures and required functionality; and

Delivering the finished application or an error message if application does not implement all logical descriptors required by the client means application request.